(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67785 A2

(51) International Patent Classification[7]: H04Q 7/00

(21) International Application Number: PCT/EP01/02455

(22) International Filing Date: 5 March 2001 (05.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0005173.0 4 March 2000 (04.03.2000) GB

(71) Applicant *(for all designated States except US)*: **MOTOROLA INC.** [US/US]; 1303 E. Algonquin Road, 3rd floor, Schaumburg, IL 60196 (US).

(71) Applicants and
(72) Inventors: **SHI, Rong** [GB/GB]; 6 Oberon Way, Abbey Meads, Swindon, Wiltshire SN2 3WH (GB). **ELLIS, Martin, John** [GB/GB]; 16 The Birches, Marlborough Road, Swindon, Wiltshire SN3 1PT (GB). **CATALDO, Mark** [GB/GB]; 5 Greenbank Crescent, Bassett, Southampton, Hampshire SO16 7FR (GB).

(74) Agents: **JEPSEN, Rene** et al.; Motorola European Intellectual Property Operations, Midpoint, Alencon Link, Basingstoke, Hampshire RG21 7PL (GB).

(81) Designated States *(national)*: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/67785 A2

(54) Title: COMMUNICATION SYSTEM ARCHITECTURE AND METHOD OF CONTROLLING DATA DOWNLOAD TO SUBSCRIBER EQUIPMENT

(57) Abstract: Code that is to be downloaded to a node (50) from across an unsecure network (12), such as the internet, is routed to a sanity check-point function (44) supporting an emulator of the node (50). Only after an interoperability and compatibility assessment has been undertaken (112, 128) by having the downloaded code run (122-126) on the emulator does the code get passed (116, 134) to the node for software upgrade, application augmentation or content review purposes. A system operator (or system administrator) can therefore regulate software upgrades to mitigate potentially catastrophic software upgrades that would otherwise adversely affect, for example, mobile unit operation. Once code has been assessed, then it can either be stored (108, 132) in a code and fault repository (46) for future use with identical node-originating requests or it can be discarded and labelled as operating-system incompatible within code and fault repository (46).

COMMUNICATION SYSTEM ARCHITECTURE AND METHOD OF CONTROLLING DATA DOWNLOAD TO SUBSCRIBER EQUIPMENT

Field of the Invention

This invention relates, in general, to a communication system controlling data download to adaptable subscriber equipment and is particularly, but not exclusively, applicable to the regulation of application software within a downloadable terminal, such as a third generation radio communication device.

Background of the Invention

With the development of communication technology, there is an ever-increasing demand by subscribers to personalise subscriber equipment functionality to meet individual (or group) requirements. Such subscriber equipment may take a number of forms, including (but not limited to): i) the so-called software radio proposed in third generation cellular systems, such as the universal system for mobile communication (UMTS); and ii) adaptable terminal equipment, such as a computer on a local area network (LAN).

For a considerable period, it has been a desire to download software. Now, with internet access and the development of data packet transfer technologies in the radio frequency domain, software download and terminal adaptation is becoming a reality. Software download from a server (or content provider) can, of course, be in a number of guises, including entire software applications (such as replacement mobile specific firmware) and software patches that address specific technical faults that have been identified subsequent to an initial release of code. Software download may also be content specific in that it is assessed on a demand basis from a content provider and may therefore appear to be general Internet information, such as e-commerce messages, web pages, etc. Furthermore, software can be provided in the form of code on an adjunct "plug-in" memory expansion cards or SIM cards for use within subscriber equipment.

In the next generation of mobile communication systems, mobile subscriber units will be able to access the Internet directly via packet switched bearers across both an air-interface and in a wireline or optical network. Furthermore, it is envisaged that services in the future will be de-coupled from the communication network, implying that the roles of network operators, service providers and manufacturers can be clearly distinguished and supported independently by unrelated parties. In theory, therefore, download of either software or content can be acquired from any accessible source. Moreover, it will be appreciated that the openness of the Internet, although desirable, leads to a very insecure network in which a subscriber can inadvertently compromise its own subscriber equipment functionality through the downloading of incompatible or deliberately malicious code. In the former instance, contemporaneous operation of downloaded code with existing software/firmware within the subscriber unit may merely inadvertently cause system failure or system crash. Maliciously produced rogue software, which is generally referred to as supporting "security attacks" may deliberately either seek to disrupt and divert data processed by the subscriber unit or, more alarmingly, to destroy third party equipment by permanently overwriting or disabling critical functions or data. Indeed, without security protection, security attacks can be relatively easily performed on specific communication packets or periods (within a call or internet access session) and to network nodes (such as subscriber units and servers).

Therefore, as regards the personalisation of subscriber equipment, there are inherent risks unfortunately associated with both the augmentation in the number of software applications supported by the subscriber equipment and the update or replacement of code from, generally, unvetted data repositories over a non-secure communication resource, such as the Internet.

To address some forms of security attack, virus checkers have been developed (for use in computer-based terminal technology) to scan for rogue software

deliberately produced to infect a host processor. Such virus checkers must be constantly updated to address new strains of computer virus.

As regards types of security attack, these can generally be classified as falling within one of two categories, namely a first category in which it is desired to seek covert access to information and a second category that operates to disrupt, adapt or otherwise compromise legitimate system or subscriber unit operation.

Security attacks in the first category include the concepts of: i) sniffing at a user request and then intercepting the software (or content) based on a perceived importance/relevance of the user request; ii) "man in the middle" access where an entity intercepts authentication requests between a user and a certification authority, thereby compromising security (in, for example, an e-commerce environment); and *iii*) general interrupt of download. The common thread between these forms of security attack is that there is direct access to the information through, for example, the tapping of a dedicated communication path. One hybrid attack mechanism relates to use of malicious code in which a third party entity (e.g. a malicious employee) gains a valid "signing certificate" by passing all certificate authority background checks, but whereafter the third party creates and signs malicious content. The first category of security attacks are difficult to address otherwise than on the basis of providing a secure communication link.

The second category of security attack goes directly to the alteration of functionality (by way of corruption of code) in the subscriber unit or host processor. There are numerous forms of such corruption, with major mechanism identified below. First, a third party may modify the software (or content) in the course of a download to a requesting subscriber unit or LAN gateway apparatus, such as a network server/node. Second, the so-called Trojan Horse attack results in a processing system appearing to action/execute requested procedure

whereas the processing system is actually doing something nefarious and supplementary to the requested procedure, e.g. control of the subscriber unit is effectively taken over by a third party unconnected with the legitimate owner. Third, an uplink message for software download is intercepted and an impersonation of an addressed server (by a third party) leads to the download of malicious or hostile code to a processor requesting the download. Fourth, a subscriber unit or server incident to downloaded software may simply be flooded with dummy codes. Fifth, a third party hacker could compromise a server to modify the software or content on the server before the start of the software downloading process.

Generally, in the event that the addressed server or data repository is not properly guarded, hackers can compromise the server by corrupting its stored data. In this case, there is no outward indication that the data offered by the server is corrupted and so, as a trusted bona fide server, the server is still accessible and interactive on the basis of authentic certificates passing between an addressing node (such as a subscriber unit) and the server itself. Clearly, in this instance, a subscriber unit may be inadvertently downloaded with corrupted software of a malicious nature. In fact, if corrupted mission-critical executable radio software (RF/IF baseband algorithms) is downloaded to a mobile unit, the mobile unit may be caused to mis-function or to terminate a function prematurely; this is of particular concern. More specifically, the problem of corrupted downloaded code in a mobile environment is exacerbated by the inherent mobile nature of the subscriber unit (e.g. a cellular phone or UMTS terminal) since contact with the mobile unit may be cut or otherwise prevented and so such contact cannot be re-established directly. Therefore, a mobile subscriber and the service provider may both be totally oblivious of any phone/equipment error or failure. As such, from a subscriber standpoint, there is simply no discernible indication of loss of service, whereas an operator will suffer loss of associated service revenue.

Clearly, it will now be appreciated that software download is very vulnerable to one or more security attacks, and a strong security mechanism is therefore preferable to safeguard the software download against security attacks in general.

The so-called Mobile Execution Environment (MexE) has been proposed in 3GPP T2 to provide automatic secure transfer of applications, applets and content. An authentication mechanism implemented in MExE is based on the CCITT X.509 digital certificate that allows a subscriber unit and server to authenticate each other effectively. A standalone encryption mechanism is used to provide privacy for downloaded software (or content). The current MExE approach for secured software/content download is to sign the software/content with a digital certificate that is authorised by a Trust Certificate Authority. The Certificate will uniquely identify the server to authenticate to the subscriber that the downloaded software/content comes from the trusted server; such a scenario exists where, for example, the server belongs to a handset manufacturer. In other words, certificates essentially contain a digital signature unique to the equipment and an encryption key for subsequent use in decoding of data packets (or the like) transferred between the subscriber unit and server.

Unfortunately, the security mechanism provided in MExE can only prevent certain security attacks, principally modification of software in the download. Besides bytecode verification that is implemented in a Java sandbox, MexE provides no explicit security protection against security attacks that directly affect an originating server or which fraudulently obtain a valid authentication certificate.

In somewhat more detail, it will be appreciated that the Java language allows Java-compatible Web browsers to download code fragments dynamically and then to execute this code in the machine. Original Java security utilises a sandbox that imposes strict controls on what Java programs can and cannot do

by limiting the rights for applets. Apart from granting Java code special privileges, Java sandbox controls access to system resources for the downloaded code.

In the context of a computer terminal/workstation, once the code is downloaded to a workstation's Java sandbox, the Java sandbox will perform serial regimes of verifications, including class file format checking and bytecode verification. In the bytecode verification, the bytecodes are executed by means of a run-time system, i.e. an emulator for the virtual machine's instruction set. A bytecode verifier verifies all the bytecodes by: i) analysing data-flows to see if there is any violation against stack overflow; and ii) accessing registers directly. The bytecode verifier is arranged to "call" (i.e. identify as being potentially corrupt) those routines or sub-routines for which, for example, inappropriate arguments are obtained or arguments are obtained with inappropriate type. However, the Java sandbox bytecode verification mechanism only performs a language dependent-type check, and acts to protect local resources from un-granted access and further prevents violation operations by ensuring that code adheres to predetermined constraints. The predetermined constraints may ensure that there are no stack overflows or underflows, and that there is valid register access and storage. Additionally, the bytecode verifier looks to the correct allocation of parameters to all bytecode instructions, and further ensures that no illegal data conversions occur. Even with this level of complexity, the Java sandbox is deficient in that it is unable to accommodate security attacks that are consequential on either a hacked compromise of software stored on a server, the use of an intercepted authentic certificate or the Trojan Horse attack.

As an intermediate summary of the Java sandbox technique, it will now be appreciated that Java sandbox provides a restrictive access function to Java code, thereby preventing access to memory locations otherwise un-associated with specific Java applets. Consequently, software interaction is limited with the Java sandbox providing an absolute security curtain.

In MExE classmark II, which is based on Personal Java, the bytecode verification is used to verify that the downloaded Java code conforms to the virtual machine execution condition. However, there is no requirement for manufacturers to utilise a single code format and so codes in languages other than Java may be used in a downloadable environment, such as in the case of executable radio system (or sub-system) software.

Of course, one could employ a brute-force mechanism of protection, whereby all forms of function personalisation are restricted. For example, computer systems have been designed to operate with a firewall/proxy/cache. A firewall/proxy can provide protection to clients by filtering/blocking attempts to retrieve content and the resulting fetched content. The mechanisms currently used by internet firewalls/proxies are very coarse (e.g. filter out all javaScript, block all accesses to site http://hackedcode.com) and cannot support the fine-grained approach needed in today's evolving communication environment.

Summary of the Invention

According to a first aspect of the present invention there is provided a communication system comprising: a third party data repository storing at least one of data content and code; and a node having means for requesting a download of at least one of the data content and code from the third party data repository; the communication system further comprising: a sanity check point function responsive to the third party data repository and coupled to the node, the sanity check point function operational to intercept at least one of data content and code downloaded from the third party data repository in response to the node requesting such download, the sanity check point function including: an emulator representative of the node and interactive with the at least one of the data content and code downloaded from the third party data repository; code assessment means for assessing an operational status of the emulator following interaction of at least one of the data content and code downloaded from the

third party data repository with the emulator representative of the node; and means for selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

In a preferred embodiment, the sanity check point function includes means for assessing at least one of compatibility of and security attacks within the data content and code, respectively, downloaded from the third party data repository.

Preferably, a code and fault repository is coupled to the sanity check point function, the code and fault repository storing at least some of: a plurality of differing node emulations associated with different forms of node coupled to the sanity check point; data content downloaded from identified third party data repositories; code downloaded from identified third party data repositories; and indications pertaining to a suitability for download to nodes of data content and code from assessments previously undertaken by the code assessment means in relation to identified third party data repositories.

In one embodiment, the sanity check point function includes: means, responsive to an address of a third party data repository generated by the means for requesting download, for searching for the address in the code and fault repository; and means for directly downloading from the code and fault repository to the node at least one of data content and code associated with the address when such is stored in the code and fault repository and subject to such stored data content and code previously yielding an operational status of the emulator satisfying the predetermined operational status.

The communication system may further comprise: a gateway to a home network containing the sanity check function; and a dedicated secure link between at least one third party data repository; and wherein the home network further

includes: means for selectively disabling the sanity check point function in response to a request to download at least one of data content and code across the dedicated secure link.

A visiting network may be coupled to the home network, with the visiting network supporting communication to a subscriber unit primarily affiliated with the home network. The communication system may then further include: means for referring a request for download from the subscriber unit in the visiting network to the sanity check point function in the home network.

A globally accessable network, such as the internet, is typically coupled between the third party data repository and the sanity check point function.

As regards the predetermined operational status, its level is arbitrarily set but may be subject to the extent to which critical operating systems are affected. Consequently, total operational compliance may be demanded in certain instances, whereas on other occasions minor faults in subsequent node operation may be acceptable (e.g. in the instance when an existing firmware problem is known within the node, such as a cell phone, the up-grade to partially operational system software may represent an improvement in operational functionality).

In a second aspect of the present invention there is provide a method of controlling data download to a node in a network of a communication system having a third party data repository storing at least one of data content and code, the method comprising: at the node, requesting a download of at least one of the data content and code from the third party data repository; the method further comprising: intercepting within the network at least one of data content and code downloaded from the third party data repository in response to the node requesting such download; generating an emulation representative of the node; causing operational interaction between the emulation and at least one of

the data content and the code downloaded from the third party data repository; assessing an operational status of the emulation following operational interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

In a further aspect of the present invention there is provided a control device for regulating download of data to a node in a network of a communication system having a third party data repository storing at least one of data content and code, the control device comprising: means operationally configured to intercept at least one of data content and code downloaded from the third party data repository in response to the node requesting such download; an emulator representative of the node and interactive with the at least one of the data content and code downloaded from the third party data repository; code assessment means for assessing an operational status of the emulator following interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and means for selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

In yet another aspect of the present invention there is provided a computer program product for a controller that controls data download to a node in a home network of a communication system having a third party data repository storing at least one of data content and code, the computer program product comprising: code, responsive to the node requesting a download, that directs the controller to intercept at least one of data content and code downloaded from the third party data repository; code that directs the controller to generate

an emulation representative of the node; code that directs the controller to cause operational interaction between the emulation and at least one of the data content and the code downloaded from the third party data repository; code that directs the controller to assess an operational status of the emulation following operational interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and code that directs the controller to selectively forward to the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status; wherein the codes reside in a computer readable medium.

Preferably, when a mobile subscriber, for example, requests download of code that could affect its operating system, such code is always sent to a sanity check-point function deterministic of code compatibility irrespective of where the mobile subscriber is located.

Advantageously, the present invention provides a mechanism by which a system operator (or other system administrator) can perform a sanity check on potentially suspect downloadable code that has been requested by a subscriber to update or augment existing code within a subscriber's node or communication device. Considering the application of the present invention to a mobile radio communication device environment, the ability of the system operator (or system administrator) to regulate software upgrade mitigates potentially catastrophic software upgrades that adversely affect mobile unit operation. More particularly, the present invention allows the system operator to veto the download of software that could otherwise disrupt and prevent both system access (in the sense of both uplink and downlink connections) to the mobile radio communication device. Whilst the present invention can strictly regulate the download of software it is, however, able to be selective in its approach to regulation based upon a level of confidence the system operator has in a subscriber-accessed site or dedicated path. For example, if the subscriber

accesses a site which in within a secure intranet, the download of software (especially application code, such as interactive Java applets) can be immediately sanctioned without administrator review and interaction. Conversely, if the subscriber accesses a site which is deemed unsecure (such as through multiple gateways into a wide area Internet domain), then a more critical stance can be adopted by the administrator whereby downloadable code is treated sceptically and therefore subjected to an integrity/compatibility assessment according to the principals of the present invention.

The present invention therefore provides a verification mechanism (that acts as a software sanity check) for software, which mechanism is actionable at a time prior to both software download and upgrade. Therefore, the present invention addresses the needs of personalisation of mobile or other subscriber units or nodes whilst protecting mission-critical software from corruption. The preferred embodiment can be implemented at a number of alternative nodes within a communication system and may therefore provide service regulation of subscribers associated with a LAN (or the like), thereby allowing the verification mechanism to be shared between multiple subscriber units or nodes.

The present invention can be used, for example, in mobile communication software download products; it can be used to enhance the security strength in MExE and the Software Defined Radio (SDR). The present invention also finds application, for example, in 3GPP TSG T2 MExE standardisation, the SDR forum and the European Union funded 5th Framework TRUST (Transparent Reconfigurable UbiquitouS Terminal) project.

Brief Description of the Drawings

An exemplary embodiment of the present invention will now be described with reference to the Drawings, in which:

FIG. 1 is a block diagram of a communication system according to a preferred embodiment of the present invention; and

FIG. 2 is a flow diagram of a preferred operating methodology of the present invention.

Detailed Description of a Preferred Embodiment

Turning to FIG. 1, there is shown a block diagram of a communication system 10 according to a preferred embodiment of the present invention. Many of the components of the communication system 10 are of conventional design.

An unsecure network 12, such as the Internet provides (typically) a packet-switched transport domain across which data traffic is passed between a home network 13 (which may be a LAN) and selective peripheral entities 14-18. By way of example and without limitation, a peripheral entity may take the form of service providers 14 containing application-based codes 20 (i.e. software and firmware) and billing information 22. Alternatively, a peripheral entity may be a content provider 16 providing services and access to e-commerce 24 or information 26 in general. A third form of peripheral entity 18 may be a server of a manufacturer, with this data site storing firmware 28 in the form of Java applets or the like.

With respect to any server 18 associated with a system manufacturer, although the site is shown to be interconnected to the home network 13 through the unsecure network 12, a supplementary or alternative connection to the server 18 could be by way of dedicated (and hence deemed secure) link 30. Generally, information transferred between the server 18 associated with a manufacturer and the home network 13 will include a certificate 32 to provide additional security (although the present invention effectively mitigates the use of certificates).

The various peripheral entities are inter-changeably known or referred to as third party data repositories, since they provide an accessible code or content resource that can be downloaded on request.

In relation to the home network 13, a gateway 34 provides a physical interface between the home network 13 and the unsecure network 12. The gateway 34 may take the form of a switch and generally provides some form of routing and/or inter-operability function, as will readily be appreciated. The gateway 34, which therefore includes control logic 35, is in turn coupled to a data network 36 (e.g. a packet-orientated domain) within the home network 13, which data network 36 allows interconnection of concerns. The gateway 34, which may be a Wireless Application Protocol (WAP) gateway, acts to pass messages in an unmolested fashion between end-points of a communication.

In a cellular-type environment, the data network 36 is typically coupled to a base station controller (BSC) 38 that administers a plurality of base station transceivers (BTSs) 40-42.

The data network 36 further provides interconnection to a sanity check-point function 44 generally realised within a processor or the like. The sanity check-point function 44, according to a preferred embodiment, is a code-based emulator representative of at least one node interconnected to the data network 36. The sanity check-point function 44 is further coupled to a code and fault repository 46 that stores emulation code as well as content and applications (e.g. software or firmware code) that have been downloaded across the unsecure network 12. The code and fault repository 46 may also include a results cache pertaining to previous emulation trails undertaken by the emulator of the sanity check-point function 44 in relation to downloaded code or content.

The data network 36 may also be a distribution point to additional visiting networks 48 to which a subscriber unit 50 (usually associated with the home network 13) can be temporarily affiliated. The subscriber unit 50 may therefore take the form of a mobile phone, for example. The construction of the visiting network 48 may reflect the home network 13 and is shown, by way of illustration,

to contain a data network (e.g. a packet-based network supporting internet protocol or the like) 52 coupled to the home network's data network 36. Additionally, data network 52 is shown to be coupled to a computer 54 via a dedicated wireline or optical connection 56. The data network 52 of the visiting network 48 acts to provide a transport mechanism for information transfers between the peripheral entities 14-18 and, say, the subscriber unit 50 via a visiting BSC 58 and associated BTSs 60-62.

The present invention provides a mechanism that safeguards principally against security attacks that directly affect code stored in peripheral entities, although the present invention has wider application. The emulator within the sanity check-point function 44 in the home network 13 operates to provide a representation of a subscriber unit requesting download of content or code from the peripheral entities 14-18. The gateway is instructed to ensure that at least selective code or content downloaded across the unsecure network 12 is initially routed to and typically thereafter stored (at least temporarily) within the code and fault repository 46. The code or content is then run on the emulator, with the compatibility and interoperability of the code and content assessed by its interaction with the resulting emulation. More specifically, the sanity check-point function 44 will support a subscriber unit emulator for each type of node (e.g. BSC, BTS or subscriber unit or computer) supported by the home network 13. The sanity check-point function 44 therefore runs hostile code checking programs to detect security attacks, such as the Trojan Horse, denial-of-service or other forms of malicious (security) code attacks hidden within the downloaded code. The emulator therefore simulates the exact functions of a real requesting subscriber entity (as previously stored in the code and fault repository 46). Consequently, only once compatibility of the code and content has been assessed does download to a requesting subscriber actually occur. If the code is deemed incompatible or harmful, then the code is purged from the emulator and the code and fault repository 46 and an indication preferably made within the code and fault repository 46 that the site of the originating code or content

should, in future, be treated sceptically or always disregarded. Such labelling of third party sites may be reviewed periodically by the home network 13 to address legitimate attempts by third party peripheral entities to address harmful or incompatible code or content.

As will be understood, the assessment of compatibility and interoperability of the code with the emulation can be based on techniques that utilise parameter checks such as arguments, overflows or underflows, and valid register access and storage within the emulation.

Preferably, the sanity check-point function 44 is owned by the home network 13 because the home network can always be assumed to be the trusted party with a well-protected network residing in the Network-Operator-Domain. As such, network nodes (such as the BSC 38, BTS 40-42 and subscriber unit 50) inside the home network are generally free of normal Internet-originating security attacks. Also, the home network 13 is usually responsible for proper operation and functioning of affiliated subscriber equipment, which means that the home network 13 can have an overall responsibility to ensure that the subscriber-specific software downloaded to a requesting subscriber unit works properly within the home network and so provides full subscriber unit functionality.

Turning to the flow diagram of FIG. 2, a preferred operating methodology for the present invention is shown. The process begins with a request 100 by a node for download of code or content from a remote third party site. Optionally, some form of certificate authentication 102 may take place to augment security. Content or code download (from the third party site) occurs 104 to the intermediate sanity check-point function 44, with this possibly requiring temporary storage of code within the code and fault repository 46. At some point in the process, the code and fault repository 46 may be checked 106-108 to identify whether an addressed (remote) third party site has been accessed before for the download of similar or identical code or content. In the affirmative

109, the system of the present invention accesses 110 memory to determine 112 whether such previously requested code or content contained no more than an acceptable level of interoperability faults (which level may have a zero-tolerance). A positive result 114 from decision block 112 results in local download 116 of code or content to the requesting node from the code and fault repository 46, thereby reducing wide area traffic. A negative result 114 from decision block 112 results in the request for download being denied and the node accordingly informed 118.

If no record of access effectively exists for the third party site, the process takes a second path 120 from decision block 108, with the second path leading to the invoking 122 of an emulator representative of the requesting node, e.g. mobile subscriber unit 50. The code or content downloaded from the third party site is run 124 with the emulator and a hostile code-checking program identifies 126-128 harmful or node incompatibility code. If the emulation is deemed unaffected or substantially unaffected (i.e. non-critically affected) by the downloaded code or content, flow proceeds along path 130 where a local record is preferably made 132 of the downloaded code (and stored for future reference in code and fault repository 46). The node requesting code or content download is then serviced 134 through the provision of downloaded data content or code. Should the decision from block 128 be affirmative 137, then the system preferably records the code but at least the existence of fault(s) in the code and fault repository 46 (i.e. a suitable database or memory device) and informs the node that download has been denied in view of incompatibility problems.

Of course, if the system (at the time of the download request) can access 140 a dedicated and hence secure path to an addressed peripheral entity, then download of the code to the requesting node may be immediate 142 and not subject to the emulation process and associated compatibility/harm assessments.

In summary, the sanity check-point functions interacts with the code and fault repository 46 to see if the code or content has been downloaded previous by other nodes within the home network 13. In the affirmative, the code and content previously stored in the code and fault repository 46 can be downloaded to the node to avoid superfluous emulation or WAN access. Otherwise, the sanity check-point function invokes the appropriate emulator for the type of node requesting download and then runs downloaded code against the node emulator. A hostile code-checking program detects possible security attacks. Once downloaded code or content passes the sanity check, it may be installed in the code and fault repository 46 for subsequent access and use by other like nodes. Conversely, if significant interoperability faults/problems are detected, the downloaded code or content (and an associated fault record) may be stored in the code and fault repository 46 in order to avoid a necessity for repeating the emulation process in response to a similar but later request from either the same node or a similar node.

It will, of course, be appreciated that the above description is given by way of example only and that modifications in detail may be made within the scope of the present invention. For example, the concept of utilising intermediate emulation equipment, typically in the form of a virtual machine (supported entirely within code), can be applied to all software or content supplied to a central node. The terms "third party data repository", "peripheral entity" or the like should therefore be considered in a broader sense to include, as necessary and when the context requires, the loading of a third party CD-ROM (or the like) into a central node for subsequent dissemination on a demand-driven basis. Consequently, inter-operability of software can be assessed by the present invention thereby yielding improved roll-out of software upgrades (arising from an ability to identify potential incompatibility/harmful code at an earlier, off-line stage). Therefore, a subscriber receiving a software upgrade can be confident that their subscriber unit is compatible with such new software and that any

subsequent fault is therefore generally likely to be attributable to a hardware malfunction.

Claims

1. A communication system comprising:

a third party data repository storing at least one of data content and code; and

a node having means for requesting a download of at least one of the data content and code from the third party data repository;

the communication system further comprising:

a sanity check point function responsive to the third party data repository and coupled to the node, the sanity check point function operational to intercept at least one of data content and code downloaded from the third party data repository in response to the node requesting such download, the sanity check point function including:

an emulator representative of the node and interactive with the at least one of the data content and code downloaded from the third party data repository;

code assessment means for assessing an operational status of the emulator following interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and

means for selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

2. The communication system according to claim 1, wherein the sanity check point function includes means for assessing at least one of compatibility of and security attacks within the data content and code, respectively, downloaded from the third party data repository.

3. The communication system according to claim 1 , further comprising a code and fault repository coupled to the sanity check point function, the code and fault repository storing at least some of:

a plurality of differing node emulations associated with different forms of node coupled to the sanity check point;

data content downloaded from identified third party data repositories;

code downloaded from identified third party data repositories; and

indications pertainiing to a suitability for download to nodes of data content and code from assessments previously undertaken by the code assessment means in relation to identified third party data repositories.

4. The communication system according to claim 3, wherein in the sanity check point function includes:

means, responsive to an address of a third party data repository generated by the means for requesting download, for searching for the address in the code and fault repository; and

means for directly downloading from the code and fault repository to the node at least one of data content and code associated with the address when such is stored in the code and fault repository and subject to such stored data content and code previously yielding an operational status of the emulator satisfying the predetermined operational status.

5. The communication system according to any claim 1, further comprising:

a gateway to a home network containing the sanity check function; and

a dedicated secure link between at least one third party data repository;

and wherein the home network further includes:

means for selectively disabling the sanity check point function in response to a request to download at least one of data content and code across the dedicated secure link.

6. The communication system according to claim 5, further comprising a visiting network coupled to the home network, the visiting network supporting communication to a subscriber unit primarily affiliated with the home network, the communication system further including:

means for referring a request for download from the subscriber unit in the visiting network to the sanity check point function in the home network.

7. The communication system according to claim 1, wherein the node is serviced by a radio communication resource.

8. The communication system according to claim 1, further comprising a globally accessable network coupled between the third party data repository and the sanity check point function.

9. The communication system according to claim 1, wherein the predetermined operational status is total operational compliance.

10. A method of controlling data download to a node in a network of a communication system having a third party data repository storing at least one of data content and code, the method comprising:

at the node, requesting a download of at least one of the data content and code from the third party data repository;

the method further comprising:

intercepting within the network at least one of data content and code downloaded from the third party data repository in response to the node requesting such download;

generating an emulation representative of the node;

causing operational interaction between the emulation and at least one of the data content and the code downloaded from the third party data repository;

assessing an operational status of the emulation following operational interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and

selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

11. The method of controlling data download according to claims 10, further comprising assessing at least one of a compatibility of and security attacks within the data content and code, respectively, downloaded from the third party data repository.

12. The method of controlling data download according to claim 10, further comprising storing, in a code and fault repository within the network, at least some of:

a plurality of differing node emulations associated with different forms of node within the communication system;

data content downloaded from identified third party data repositories;

code downloaded from identified third party data repositories; and

indications pertaining to a suitability for download to nodes of data content and code from assessments previously undertaken by the code assessment means in relation to identified third party data repositories.

13. The method of controlling data download according to claim 12, further comprising:

searching for an address of a third party data repository in a code and fault repository of the network; and

downloading directly from the code and fault repository to the node at least one of data content and code associated with the address when such is stored in the code and fault repository and subject to such stored data content

and code previously yielding an operational status of the emulator satisfying the predetermined operational status.

14. The method of controlling data download according to claim 10 , wherein the network further includes a gateway coupled to at least one third party data repository through a dedicated secure link, the method further comprising;

selectively disabling the generation of the emulation and the assessing of the operation status in response to the request to download at least one of data content and code across the dedicated secure link.

15. The method of controlling data download according to claim 14, further referring a request for download from a subscriber unit in a visiting network via a home network.

16. The method of controlling data download according to claim 10 , wherein the node is serviced by a radio communication resource.

17. The method of controlling data download according to claim 10 , wherein a globally accessable network couples the third party data repository to the home network.

18. A control device for regulating download of data to a node in a network of a communication system having a third party data repository storing at least one of data content and code, the control device comprising:

means operationally configured to intercept at least one of data content and code downloaded from the third party data repository in response to the node requesting such download;

an emulator representative of the node and interactive with the at least one of the data content and code downloaded from the third party data repository;

code assessment means for assessing an operational status of the emulator following interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and

means for selectively forwarding to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status.

19. The control device according to claim 18, further comprising means for assessing at least one of compatibility of and security attacks within the data content and code, respectively, downloaded from the third party data repository.

20. The control device according to claim 18 , coupled to a code and fault repository storing at least some of:

a plurality of differing node emulations associated with different forms of node coupled to the sanity check point;

data content downloaded from identified third party data repositories;

code downloaded from identified third party data repositories; and

indications pertaining to a suitability for download to nodes of data content and code from assessments previously undertaken by the code assessment means in relation to identified third party data repositories.

21. The control device according to claim 20, further including:

means, responsive to an address of a third party data repository generated by the node, for searching for the address in the code and fault repository; and

means for directly downloading from the code and fault repository to the node at least one of data content and code associated with the address when such is stored in the code and fault repository and subject to the stored data

content and code previously yielding an operational status of the emulator satisfying the predetermined operational status.

22. The control device according to claim 18 , further comprising:

means for selectively disabling the emulator in response to a request to download at least one of data content and code across a dedicated secure link incident to the control device.

23. A computer program element comprising computer program code means for making a controller execute procedure to perform the method steps of claim 10.

24. The computer program product of claim 23, embodied on a computer readable medium.

25. A computer program product for a controller that controls data download to a node in a home network of a communication system having a third party data repository storing at least one of data content and code, the computer program product comprising:

code, responsive to the node requesting a download, that directs the controller to intercept at least one of data content and code downloaded from the third party data repository;

code that directs the controller to generate an emulation representative of the node;

code that directs the controller to cause operational interaction between the emulation and at least one of the data content and the code downloaded from the third party data repository;

code that directs the controller to assess an operational status of the emulation following operational interaction of at least one of the data content and code downloaded from the third party data repository with the emulator representative of the node; and

code that directs the controller to selectively forward to the node the at least one of the data content and code downloaded from the third party data repository subject to the assessment of the operation status satisfying a predetermined operational status;

wherein the codes reside in a computer readable medium.

26. The computer program product of claim 25, further comprising:

code that directs the controller to assess at least one of a compatibility of and security attacks within the data content and code, respectively, downloaded from the third party data repository.

27. The computer program product of claim 25, further comprising:

code that directs the controller to search for an address of a third party data repository in a code and fault repository of a home network; and

code that directs the controller to download directly from the code and fault repository to the node at least one of data content and code associated with the address when such is stored in the code and fault repository and subject to such stored data content and code previously yielding an operational status of the emulator satisfying the predetermined operational status.

28. The computer program product of claim 25, wherein the network further includes a gateway coupled to at least one third party data repository through a dedicated secure link, and wherein the computer program product further comprises;

code that directs the controller to selectively disable the generation of the emulation and the assessing of the operation status in response to the request to download at least one of data content and code across the dedicated secure link.

29. The computer program product of claims 25, further comprising:

code that directs the controller to refer a request for download from a subscriber unit in a visiting network via the subscriber unit's home network.

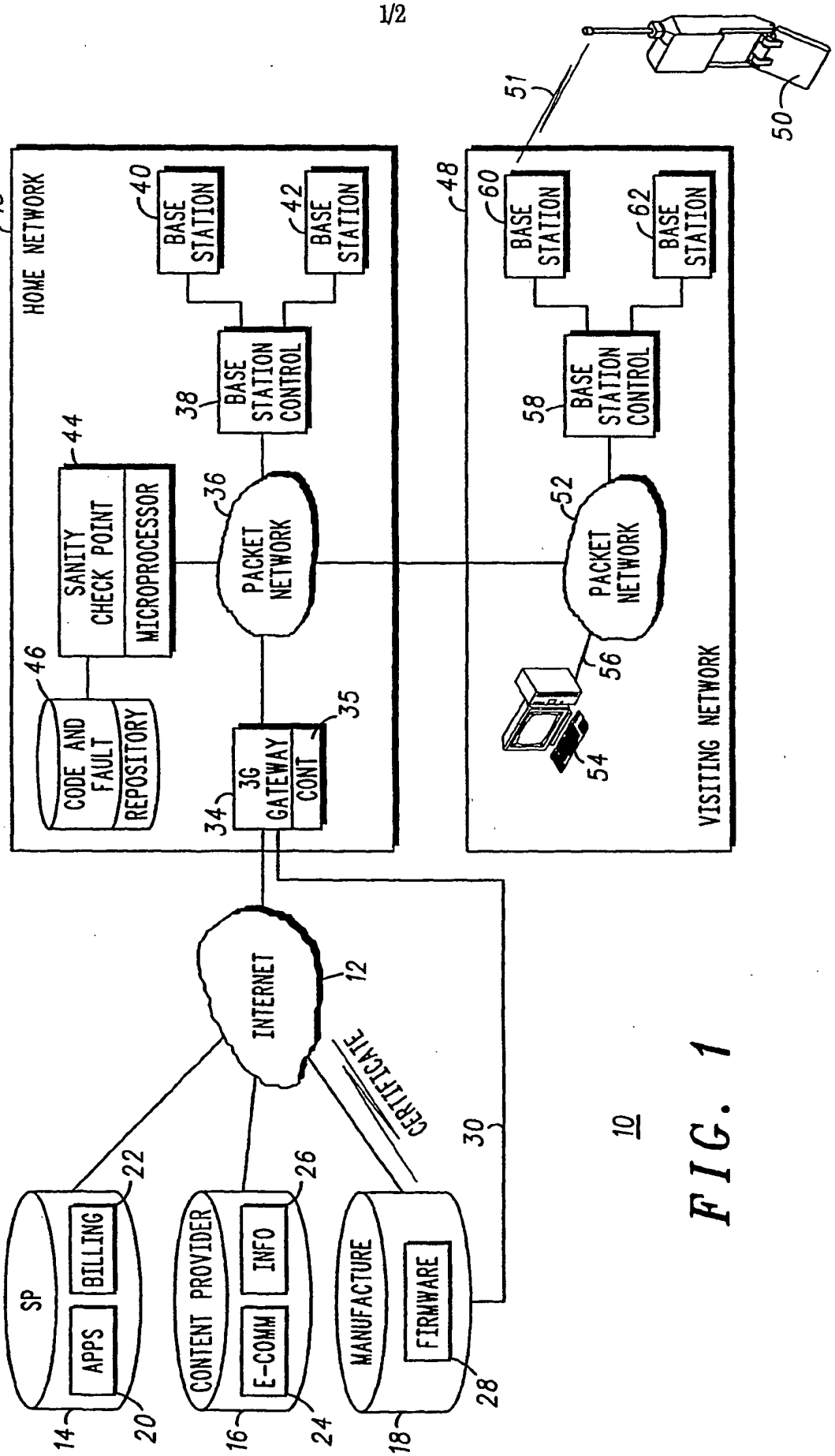

HOME NETWORK
13
BASE STATION
40
BASE STATION
42
BASE STATION CONTROL
38
SANITY CHECK POINT
MICROPROCESSOR
44
PACKET NETWORK
36
CODE AND FAULT REPOSITORY
46
3G GATEWAY CONT
35
34
VISITING NETWORK
48
BASE STATION
60
BASE STATION
62
BASE STATION CONTROL
58
PACKET NETWORK
52
56
54
51
50
INTERNET
12
CERTIFICATE
30
SP
APPS
BILLING
22
20
14
CONTENT PROVIDER
E-COMM
INFO
26
24
16
MANUFACTURE
FIRMWARE
28
18
10

FIG. 1

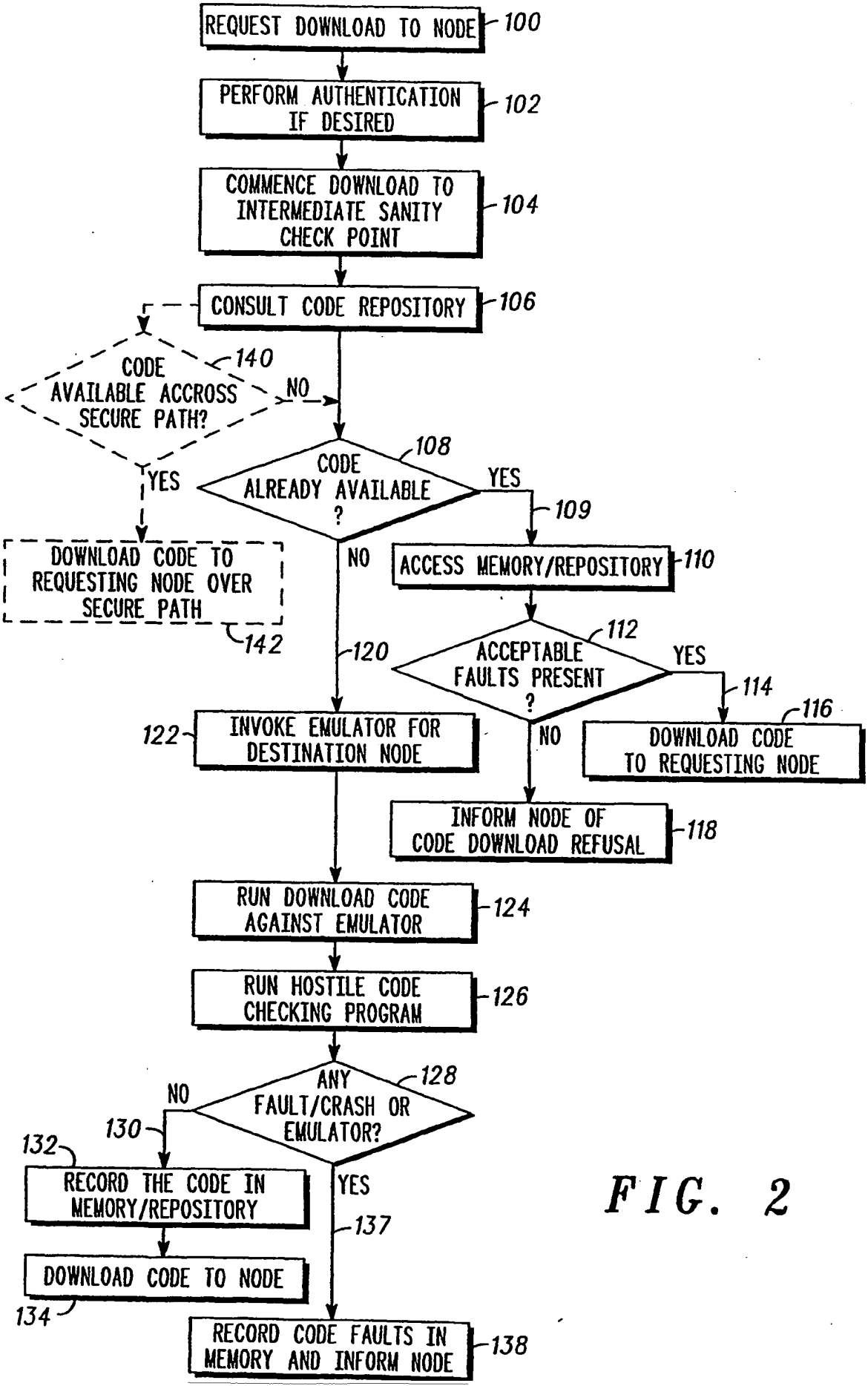

REQUEST DOWNLOAD TO NODE
100
PERFORM AUTHENTICATION IF DESIRED
102
COMMENCE DOWNLOAD TO INTERMEDIATE SANITY CHECK POINT
104
CONSULT CODE REPOSITORY
106
CODE AVAILABLE ACCROSS SECURE PATH?
140
NO
YES
CODE ALREADY AVAILABLE ?
108
YES
109
NO
DOWNLOAD CODE TO REQUESTING NODE OVER SECURE PATH
142
ACCESS MEMORY/REPOSITORY
110
ACCEPTABLE FAULTS PRESENT ?
112
YES
114
NO
120
DOWNLOAD CODE TO REQUESTING NODE
116
INVOKE EMULATOR FOR DESTINATION NODE
122
INFORM NODE OF CODE DOWNLOAD REFUSAL
118
RUN DOWNLOAD CODE AGAINST EMULATOR
124
RUN HOSTILE CODE CHECKING PROGRAM
126
ANY FAULT/CRASH OR EMULATOR?
128
NO
130
YES
132
RECORD THE CODE IN MEMORY/REPOSITORY
137
DOWNLOAD CODE TO NODE
134
RECORD CODE FAULTS IN MEMORY AND INFORM NODE
138

FIG. 2

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67785 A3

(51) International Patent Classification[7]: H04Q 7/32, H04L 9/00, 29/06

(21) International Application Number: PCT/EP01/02455

(22) International Filing Date: 5 March 2001 (05.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0005173.0 4 March 2000 (04.03.2000) GB

(71) Applicant *(for all designated States except US)*: **MOTOROLA INC.** [US/US]; 1303 E. Algonquin Road, 3rd floor, Schaumburg, IL 60196 (US).

(71) Applicants and
(72) Inventors: **SHI, Rong** [GB/GB]; 6 Oberon Way, Abbey Meads, Swindon, Wiltshire SN2 3WH (GB). **ELLIS, Martin, John** [GB/GB]; 16 The Birches, Marlborough Road, Swindon, Wiltshire SN3 1PT (GB). **CATALDO, Mark** [GB/GB]; 5 Greenbank Crescent, Bassett, Southampton, Hampshire SO16 7FR (GB).

(74) Agents: **JEPSEN, Rene** et al.; Motorola European Intellectual Property Operations, Midpoint, Alencon Link, Basingstoke, Hampshire RG21 7PL (GB).

(81) Designated States *(national)*: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW.

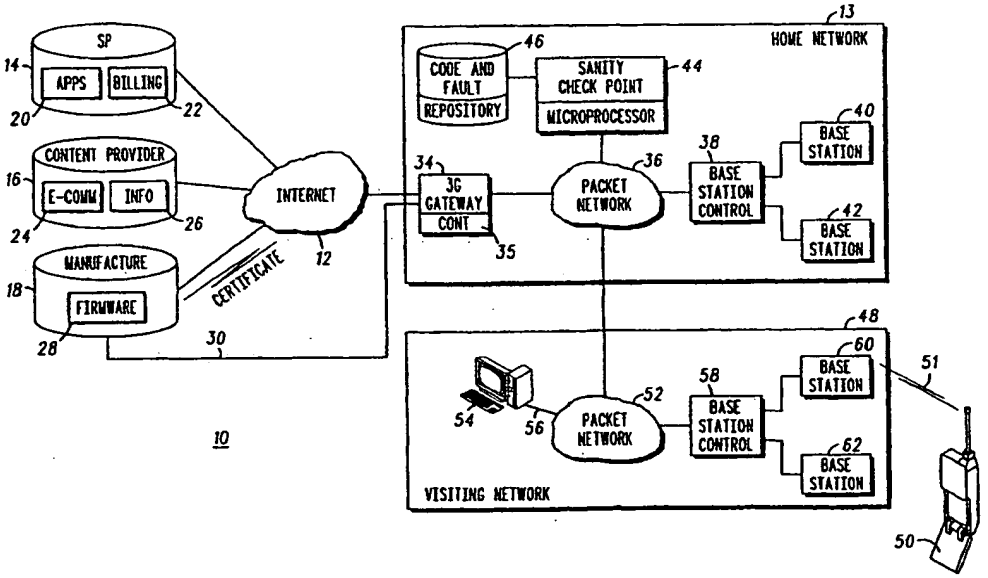
(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*

(88) Date of publication of the international search report:
27 December 2001

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SECURE DATA DOWNLOAD



(57) Abstract: Code that is to be downloaded to a node (50) from across an unsecure network (12), such as the internet, is routed to a sanity check-point function (44) supporting an emulator of the node (50). Only after an interoperability and compatibility assessment has been undertaken (112, 128) by having the downloaded code run (122-126) on the emulator does the code get passed (116, 134) to the node for software upgrade, application augmentation or content review purposes. A system operator (or system administrator) can therefore regulate software upgrades to mitigate potentially catastrophic software upgrades that would otherwise adversely affect, for example, mobile unit operation. Once code has been assessed, then it can either be stored (108, 132) in a code and fault repository (46) for future use with identical node-originating requests or it can be discarded and labelled as operating-system incompatible within code and fault repository (46).

WO 01/67785 A3

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/EP 01/02455

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04Q7/32 H04L9/00 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04Q H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 95 33237 A (QUANTUM LEAP INNOVATIONS INC) 7 December 1995 (1995-12-07)	1,2, 7-11, 16-19, 23-26,29
	abstract	
A	page 5, paragraph 3	3-6, 12-15, 20-22, 27,28
	page 10, line 10 -page 11, line 20	
	page 12, line 16 -page 13, line 9	
	page 14, line 14 - line 26	

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 August 2001	03/09/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Blanco Cardona, P

1

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/EP 01/02455

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
P,X	WO 00 74412 A (ERICSSON TELEFON AB L M) 7 December 2000 (2000-12-07) abstract page 1, line 21 -page 3, line 10 page 6, paragraph 3	1,2, 7-11, 16-19, 23-26,29
A	WO 98 21683 A (FINJAN SOFTWARE LTD) 22 May 1998 (1998-05-22) abstract	1,3,4, 10,12, 13,18, 20,21, 25,27
A	WO 97 05551 A (VERIFONE INC ;CARLOGANU MARIUS M (FR); SHEETS JOHN F (FR)) 13 February 1997 (1997-02-13) abstract page 10, paragraph 4 page 13, paragraph 2 page 34, paragraph 4	5,6,14, 15,22,28

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9533237 A	07-12-1995	AT 183592 T	15-09-1999
		CA 2191205 A	07-12-1995
		DE 69511556 D	23-09-1999
		EP 0769170 A	23-04-1997
		JP 10501354 T	03-02-1998
		US 5842002 A	24-11-1998
WO 0074412 A	07-12-2000	AU 4968800 A	18-12-2000
		SE 9901904 A	27-11-2000
WO 9821683 A	22-05-1998	EP 0965094 A	22-12-1999
		US 6167520 A	26-12-2000
		US 6092194 A	18-07-2000
		US 6154844 A	28-11-2000
WO 9705551 A	13-02-1997	AU 730253 B	01-03-2001
		AU 6547696 A	26-02-1997
		EP 0842471 A	20-05-1998
		US 6226749 B	01-05-2001